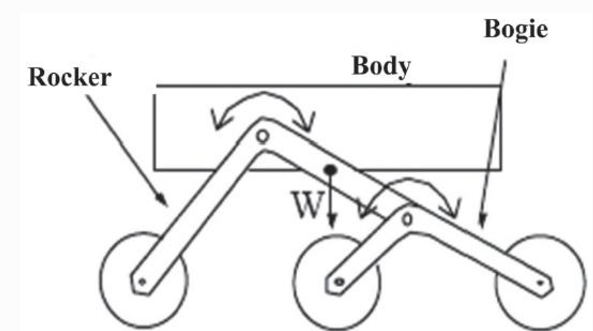
**Galaxy Rover Lessons**

**Lesson 1: General Rovers**

* Mars rovers have been built with the purpose to explore Mars by taking pictures, video, mineral samples, to find if there has ever or continues to be water, and oxygen experiments – all with the ultimate goal to survey the land before a manned mission
* Mars rovers have been sent over to the red planet starting in 1997 with **Sojourner** which lasted 93 days
* **Spirit and Opportunity** both launched in 2004 and sent to different places on Mars with more advanced cameras
* **Curiosity** launched in 2012 before **Perseverance** in 2021 and both are still operational on Mars today

**Lesson 2: Understanding the Rocker-Bogie System**

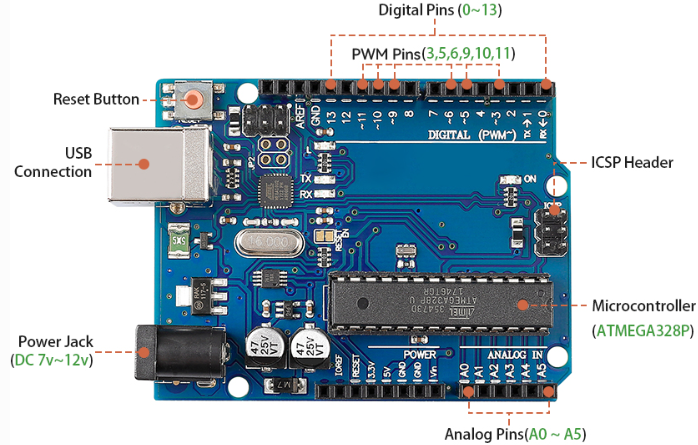
* All Mars rovers, even with upgrades to tech and capability, all kept the **Rocker-Bogie suspension** system
* Main purpose – keep all wheels (6) on the ground no matter the terrain



* The **Rocker** part is the main arm attached to the wheels and to the chassis through a differential mechanism
* As the rocker “rocks” in opposite directions to the body, the body maintains the average angle of the rockers
* The **Bogie** has similar function to the rocker, but is smaller and is attached at one end of the main rocker rather than directly to the chassis
* The main advantage is that this suspension keeps the chassis relatively level and stable, this is most important with all the technology and equipment on these rovers
* **Self-locking nuts** contain a rubber ring inside a regular nut that ensures parts won’t easily loosen due to vibration. It also gives parts extra range of motion without coming loose.

**Lesson 3: Arduino**

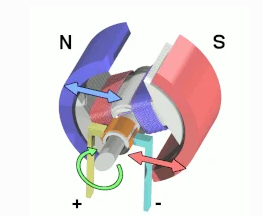
* **Arduino** is an open-sourced electronics platform, designed to make digital devices interactive and sense the physical world around them
* **Microcontroller** – a tiny computer that can run simple software
* Development board – if the microcontroller is the brain, the development board is like the body. They contain parts to help with the micro controller like oscillators for timing, voltage regulators to control power levels, and connectors for power and data flow.
* Getting to know your Sunfounder R3 board



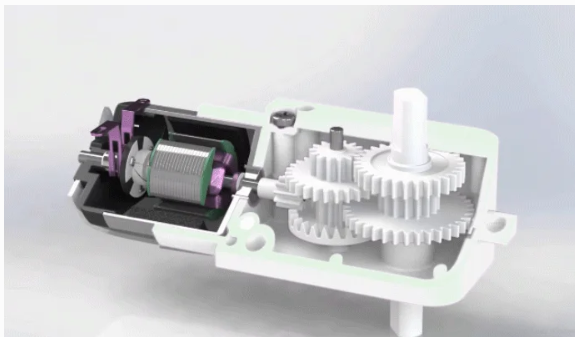
* **Digital Pins** – programmed for binary, send (output) or receive (input), yes or no, 1 or 0, on or off
* **Analog Pins** – these pins are more used to read signals on a range of values rather than 1 or 0 such as temperature, brightness (or dimness) level.
* ICSP Header – can be used for inputting special programming
* Basic Set up of Arduino Code: Setup() and Loop()
* **Setup** is usually first programmed and only happens once when we run our code. This is where we usually set our pin modes, variables, and upload libraries
* **Loop** is where we actually tell what actions our variables, outputs, sensors etc. are going to be doing
* Semi colons are read as full stop
* Curly braces mark the beginning and end of a code block
* **Functions**
  + pinMode() – decides whether a pin will be an input or an output
  + digitalWrite() – giving the action to a pin
  + Comments - // single line or /\* \*/ for multiple lines

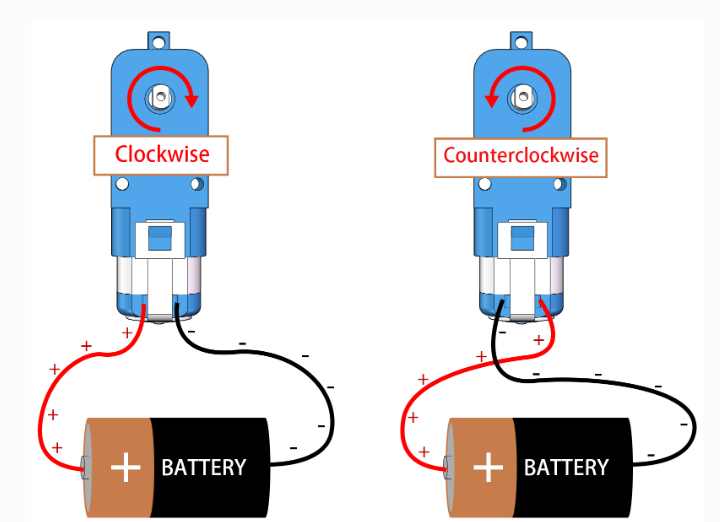
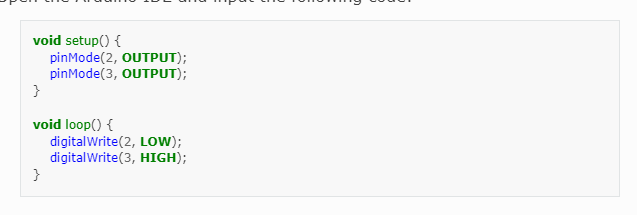
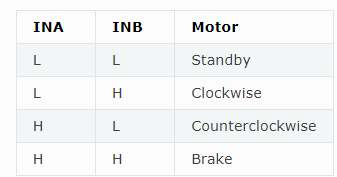
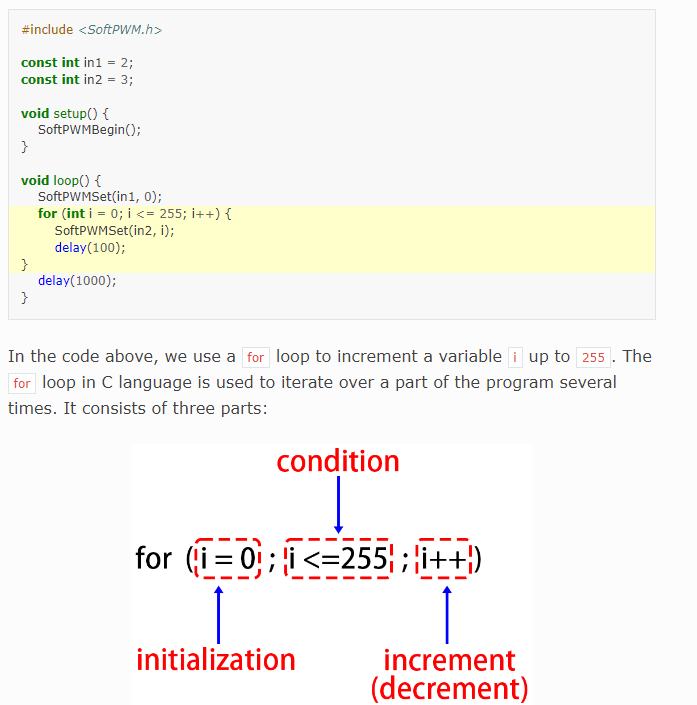
**Lesson 4: Mastering the TT Motor**

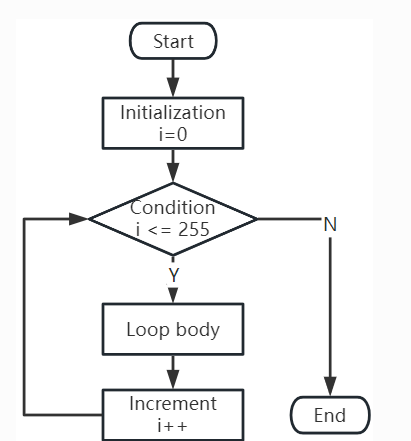
* How the motor works: the principles of **electromagnetic induction**
  + When electricity is supplied to a motor, it generates a magnetic field that then interacts with the other magnetic fields in the motor, causing the motor to spin



* The type of motor we are using in the rover is called a TT gear motor – as the motor spins (from above) it interacts with the gears enabling it to move our wheels



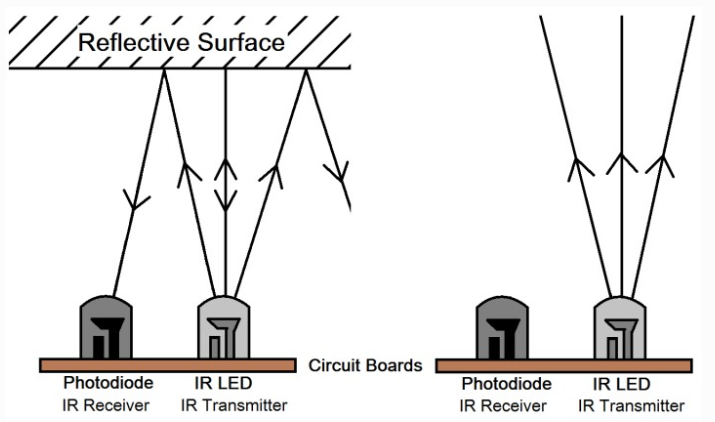
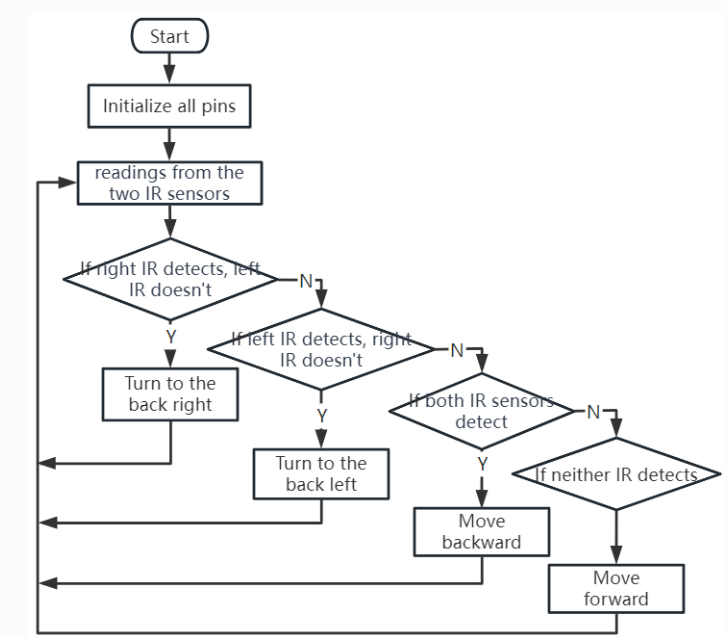
* Function and Operation
  + Example of connecting the motor right to a battery – the motor spins from the direction of the positive charge
  + 
  + We want to control the spin of the motor, this is why we connect the motor to the Arduino board (through a motor driver to amplify the current) and then the battery rather than right to the battery
* Code for rotation
  + The motor drivers control the power output to move the motor one direction of another
  + 
  + This will move the motor clock wise essentially turning on rotator on (HIGH) and the other off (LOW) – switching these will have the motor turn the opposite way
  + Logic: 
* If we just use HIGH and LOW that is the equivalent of flooring the pedal and then not pressing it at all – to control the speed we want to use a concept called **Pulse Width Modulation PWM**
  + **PWM** works by rapidly switching between HIGH and LOW digital in order to imitate an analog output
  + Using this technique with a specified pin if we have other pins currently used as analogs
* 
* Initialization – executed first and only once when we enter the FOR loop for the first time – this one tells us to start at 0
* Condition – Next step after initializing
  + If the condition is TRUE, the body of the loop is executed
* Ex – Start at 0, if i is less than or equal to 255, then add 1 to i and keep increasing by 1 as we run through the loop each time
* This code will gradually increase the speed of the motor after every iteration the code runs through the loop



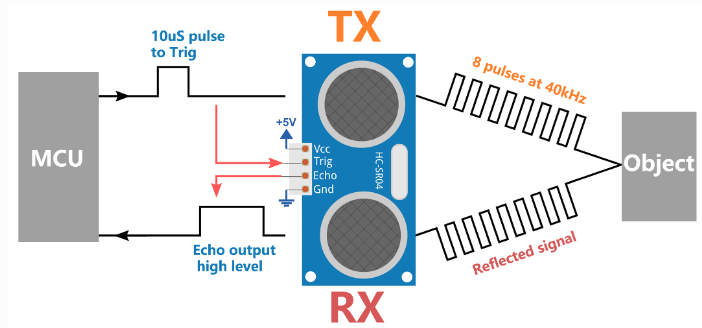
**Lesson 5: Mars Rover Mobility**

* When a vehicle is moving, the right wheels are moving clockwise and the left wheels are moving counterclock wise so to move in the same direction, we have to tell the wheels to move different directions if they are on different sides of the vehicle
* SoftPWMSet() is the function used to define our motor pins
* When defining the constants – note that each pin does not correspond to a specific motor, but to a direction of all the motors on one side, left or right
* NOTE: PRESSING THE RESET BUTTON DOES NOT RESET OR REMOVE THE CODE – IT SIMPLY TELLS THE CODE TO STOP AND RERUN FROM THE BEGINNING

**Lesson 6: Infrared Obstacle Avoidance Module**

* How the module works
  + It uses a pair of infrared components called a transmitter and a receiver
  + The transmitter acts like a flashlight emitting the infrared light and when an obstacle appears, the light bounces back and is caught by the receiver
* 
* The module can detect objects 2-40cm away
* Darker objects are harder to detect and usually detected at a shorter range, white walls get detected the furthest as an example
* The EN pin’s low-level state activates the module with a jumper cap securing the EN pin to GND
  + This jumper cap needs to be removed if we want to control the EN pin with code
* There are also 2 potentiometers, one for adjusting power, and the other for adjusting transmitting frequency
* 
* These modules are using pins 7 and 8
* Obstacle Avoidance System
* 

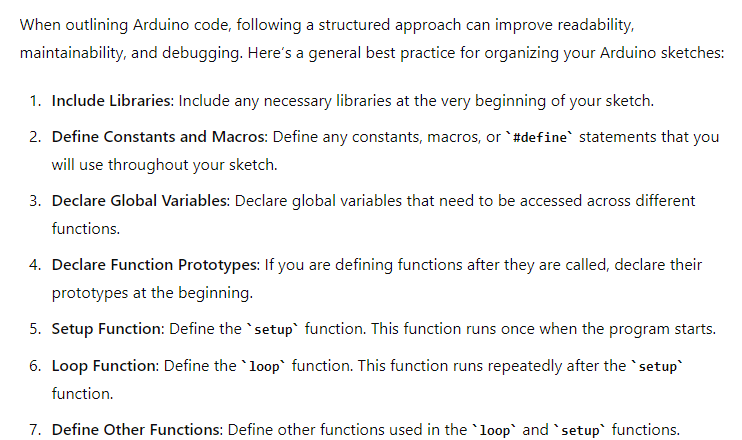
**Lesson 7 and 8: Ultra Sonic Navigation**

* The ultrasonic distance sensor can act as a detection module for objects directly in front of the rover
* This module uses sound waves similar to a bat’s sonar, to pick up waves 2cm to 400cm away
* Trigger Pulse Input – sends out the sonic waves
* Echo Pulse Output – listens to the echo to the sound waves sent out
* 
* From the microcontroller, a electric pulse is sent out to the TX which then sends out the ultra-sonic pulses at 40k hz
* If an object is present, the reflected signal travels back to the module, picked up by RX and then acts as the input sent to the microcontroller
* If an echo is returned, then a time is taken between when a signal was sent out and how long it took to return and be read – this will tell us how far away the object is
* Formula: (Time between signal sent and returned / 2) \* 340m/s
  + Which is the speed of sound 340m/s
* Our sonar module is connected to Pin 10

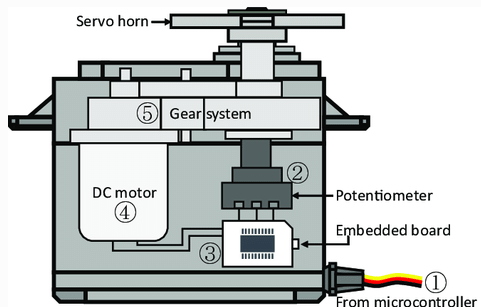
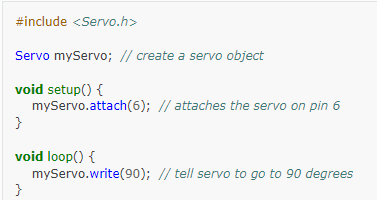
**Lesson 9: RGB LED**

* Positive end is known as the anode
* Negative end known as the cathode
* Each of our 5050 LEDs contain 3 bulbs, red, green, and blue that mix together to form most any color
* Note that each strip of our LEDs is controlled by the same set of pins so each strip acts the same as the other

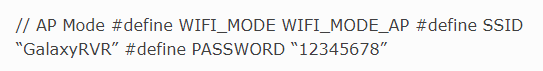
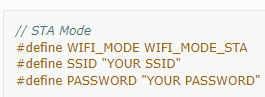
Side Note: General Arduino Template:



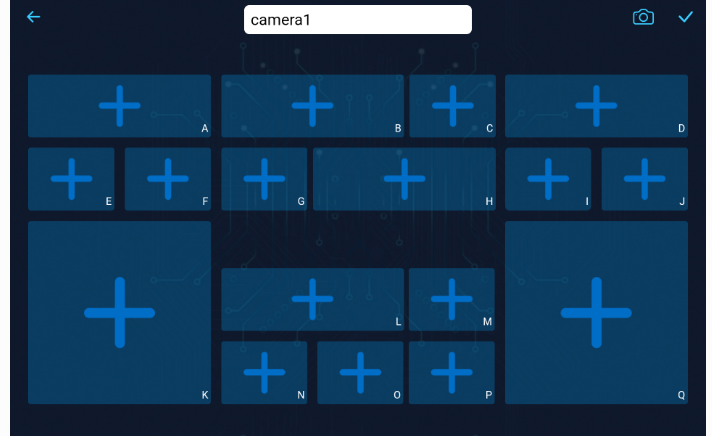
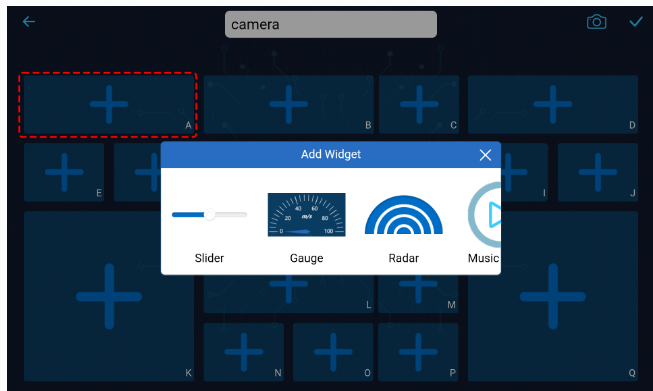
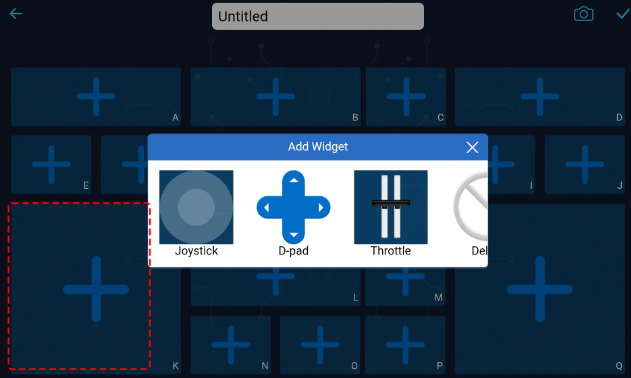
**Lesson 10: Visual Systems – Tilt and Servo Mechanism**

* Servo is a specialized motor that can move and hold specific positions (unlike a motor that just spins)
* It has a large gear around the motor which is connected to a smaller gear to make the movement slower and more powerful
* ****
* The potentiometer turns with the servo and measures what position the servo is in
* Arduino comes with a built-in library for Servos – all we need to do is assign the pin and write() to set the angle
* ****

**Lesson 11: Camera and Real-time Control**

* ESP32 Cam – compact camera module with wifi sync capabilities used to send visual data to smart phones or computers in real-time
* AP Mode
  + Rover creates a hotspot called GalaxyRVR
  + Allows devices like tablets or phones to connect and control the rover under any circumstances
  + Trade off is that the device that is connected to the Rover hotspot will not be able to connect the internet while in use with the Rover
  + 
* STA Mode
  + Rover connects to your home wifi network
  + The device used to control the Rover also needs to be connected to that same wifi network
  + Trade off is that this is limited for control only within the home wifi range
  + 
* Every time a code change is made:
  + Flip L switch to UPLOAD
  + Upload the code from Arduino
  + Flip the L switch out of UPLOAD (back)

**Lesson 12: Driving the Rover with the App**

* When opening the app, we can customize our interface by clicking on a blank space and adding the type of widget
* 
* 
* 
* Two types of widgets:
  + Control – through the app you control the movements of the rover live move forward, tilt camera, etc.
  + Show – allows you to see specific values such as tilt angle, other sensor values
* Each section has a region identifier A-Q so that we can interact with our code by assigning the region as a variable. Such as region B corresponds to the camera servo

**Lesson 13: Rover Power System**

* Batteries act as energy storage and used when needed and deployed to the specified parts to execute the task
* Our solar panels act as a way to replenish the power used by in the battery
* The solar panels are made of material that converts light into electricity – a process called photovoltiac